

Interview Summary	Application No.	Applicant(s)	
	10/607,593	FATHALLA, DIAA	
	Examiner	Art Unit	
	Tuan A. Vu	2193	

All participants (applicant, applicant's representative, PTO personnel):

(1) Tuan A. Vu. (3)_____.

(2) Mauricio Uribe. (4)_____.

Date of Interview: 17 August 2007.

Type: a) ☒ Telephonic b) ☐ Video Conference
c) ☐ Personal [copy given to: 1) ☐ applicant 2) ☐ applicant's representative]

Exhibit shown or demonstration conducted: d) ☐ Yes e) ☒ No.
If Yes, brief description: _____.

Claim(s) discussed: 1, 11 and 19.

Identification of prior art discussed: n/a.

Agreement with respect to the claims f) ☒ was reached. g) ☐ was not reached. h) ☐ N/A.

Substance of Interview including description of the general nature of what was agreed to if an agreement was reached, or any other comments: See Continuation Sheet.

(A fuller description, if necessary, and a copy of the amendments which the examiner agreed would render the claims allowable, if available, must be attached. Also, where no copy of the amendments that would render the claims allowable is available, a summary thereof must be attached.)

THE FORMAL WRITTEN REPLY TO THE LAST OFFICE ACTION MUST INCLUDE THE SUBSTANCE OF THE INTERVIEW. (See MPEP Section 713.04). If a reply to the last Office action has already been filed, APPLICANT IS GIVEN A NON-EXTENDABLE PERIOD OF THE LONGER OF ONE MONTH OR THIRTY DAYS FROM THIS INTERVIEW DATE, OR THE MAILING DATE OF THIS INTERVIEW SUMMARY FORM, WHICHEVER IS LATER, TO FILE A STATEMENT OF THE SUBSTANCE OF THE INTERVIEW. See Summary of Record of Interview requirements on reverse side or on attached sheet.



Examiner Note: You must sign this form unless it is an Attachment to a signed Office action.

Examiner's signature, if required

Continuation of Substance of Interview including description of the general nature of what was agreed to if an agreement was reached, or any other comments: The Applicant's representative was contacted in order for the Examiner to propose some additional teaching to be included to the replay tool claimed in claims 1, 11, 19. In order to render the claims non-obviousness over the prior art of record, the changes as proposed comprise (i) clarifying of the effect of the use of table to collect symbols and memory references extracted from some specific log recording previously invoked APIs calls, and (ii) based on these logged APIs using a execution template to correlate the associated binary files and their resources, such that the files and the resources are included in a API call build instance allocated for a execution environment. As agreed upon, when the changes are included, an examiner's amendment would be effectuated .

DRAFT -- Proposed Claim Amendments allowing Examiner's Amendment

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicants: D. Fathalla Attorney Docket No.: MSFT125572
Application No.: 10/607,593 Art Unit: 2193 / Confirmation No: 6013
Filed: June 27, 2003 Examiner: T. Vu
Title: APPLICATION PROGRAM INTERFACE CALL REPLAY TOOL

AMENDMENT B

Seattle, Washington 98101

August 18, 2007

TO THE COMMISSIONER FOR PATENTS:

INTRODUCTORY COMMENTS

Please amend the above-identified patent application as follows:

AMENDMENTS TO THE CLAIMS

1. (Currently amended) A computing device containing a software tool for submitting API calls corresponding to logged API calls to various components of the computing device, the software tool comprising:

symbol tables for mapping symbols, parameters and memory references ~~[[within logged]]~~ collected from a log of API calls [[with]] to corresponding memory references ~~within returns of~~ ~~logged API calls submitted by~~ allocated to the software tool, wherein the mapped symbols, parameters and memory references are used for submitting subsequent API calls corresponding to the logged API calls from within the memory space of the software tool to the various components of the computing device, and wherein the log of API calls records API calls that have been previously invoked in an execution environment of the software tool; [[and]]

an API call builder for creating an API call code sequence for submitting subsequent API calls corresponding to logged API calls to the various components of the computing device according to the mapped memory references within a symbol table; and

an execution template corresponding to binary files associated with the logged API calls for maintaining a set of resources from the binary files, wherein the execution template is utilized by an API call executor so that the API call sequence created by the API call builder is executed within the execution environment of the software tool.

2. (Previously presented) The computing device containing the software tool of claim 1 further comprising:

a memory manager that maintains memory space allocated to the software tool to simulate processes and threads associated with particular execution contexts for the logged API calls.

3. (Previously presented) The computing device containing the software tool of claim 2 further comprising:

a thread handler for establishing a thread memory structure for allocating memory and resources of the computing device, by the memory manager, that corresponds to each thread identified in the logged API calls.

4. (Canceled)

5. (Currently amended) The computing device containing the software tool of claim [[4]] 1 wherein an API call executer issues the code sequence within [[a binary]] the execution environment, including the set of resources established by the execution template.

6. (Previously presented) The computing device containing the software tool of claim 1 further comprising a replay engine for coordinating the operation of a set of handlers within the software tool to render the subsequent API calls to appropriate components of the computing device within a context defined by the software tool.

7. (Previously presented) The computing device containing the software tool of claim 6 wherein the set of handlers includes a callback handler for providing a callback destination for a callback function associated with the logged API.

8. (Previously presented) The computing device containing the software tool of claim 1 wherein the memory references comprise pointers.

9. (Previously presented) The computing device containing the software tool of claim 1 wherein the memory references comprise variables.

10. (Previously presented) The computing device containing the software tool of claim 1 wherein the code sequence comprises assembly code instructions.

11. (Currently amended) A method using a software tool for invoking API calls within an execution environment based on logged API calls, the method comprising:

~~mapping symbols, parameters and memory references in at least one symbol table within~~
~~logged API calls with to corresponding memory references within returns of logged API calls~~
~~invoked by a~~ allocated to the software tool, wherein the mapped symbols, parameters and
memory references in the at least one symbol table are collected from a log of API calls and are
used for invoking subsequent API calls based on the logged API calls from within the memory
space of the software tool, and wherein the log of API calls records API calls that have been
previously invoked in the execution environment of the software tool; and

creating an API call code sequence for invoking the subsequent API calls based on the
logged API calls, wherein the mapped memory references are specified for invoking the
subsequent API calls; and

maintaining, by an execution template corresponding to binary files associated with the
logged API calls, a set of resources from the binary files, wherein the execution template is
utilized by an API call executor so that the created API call sequence is executed within the
execution environment of the software tool.

12. (Previously presented) The method of claim 11 further comprising:

maintaining, by a memory manager, memory space allocated to the software tool for
simulating processes and threads associated with particular execution contexts for the logged
API calls.

13. (Previously presented) The method of claim 12 further comprising:

establishing, by the memory manager, a thread memory structure for allocating memory
and resources corresponding to each thread identified in the logged API calls.

14. (Canceled)

15. (Currently amended) The method of claim ~~[[14]]~~ 11 further comprising:

issuing, by an API call executor, the API call code sequence within ~~[[a binary]]~~ the
execution environment, including the set of resources established by the execution template.

16. (Original) The method of claim 11 wherein the memory references comprise pointers.

17. (Original) The method of claim 11 wherein the memory references comprise variables.

18. (Original) The method of claim 11 wherein the API call code sequence comprises assembly code instructions.

19. (Currently amended) A computer-readable storage medium including computer-executable instructions for invoking API calls within an execution environment of a software tool based on logged API calls to various components of a computing device, the computer-executable instructions performing the steps of:

mapping symbols, parameters and memory references in at least one symbol table within logged API calls with to corresponding memory references within returns of logged API calls invoked by a allocated to the software tool, wherein the mapped symbols, parameters and memory references in the at least one symbol table are collected from a log of API calls and are used for invoking subsequent API calls based on the logged API calls from within the memory space of the software tool, and wherein the log of API calls records API calls that have been previously invoked in the execution environment of the software tool; and

creating an API call code sequence for invoking the subsequent API calls based on the logged API calls, wherein the mapped memory references are specified for invoking the subsequent API calls; and

maintaining, by an execution template corresponding to binary files associated with the logged API calls, a set of resources from the binary files, wherein the execution template is utilized by an API call executor so that the created API call sequence is executed within the execution environment of the software tool.

20. (Previously presented) The computer-readable storage medium of claim 19 further comprising computer-executable instructions performing the steps of:

maintaining, by a memory manager, memory space allocated to the software tool for simulating processes and threads associated with particular execution contexts for the logged API calls.

21. (Previously presented) The computer-readable storage medium of claim 20 further comprising computer-executable instructions performing the steps of:

establishing, by the memory manager, a thread memory structure for allocating memory and resources corresponding to each thread identified in the logged API calls.

22. (Canceled)

23. (Currently amended) The computer-readable storage medium of claim ~~[[22]]~~ 19 further comprising computer-executable instructions performing the steps of:

issuing, by an API call executer, the API call code sequence within ~~[[a binary]]~~ the execution environment, including the set of resources established by the execution template.

24. (Previously presented) The computer-readable storage medium of claim 19 wherein the memory references comprise pointers.

25. (Previously presented) The computer-readable storage medium of claim 19 wherein the memory references comprise variables.

26. (Previously presented) The computer-readable storage medium of claim 19 wherein the code sequence comprises assembly code instructions.

REMARKS

Respectfully submitted,

CHRISTENSEN O'CONNOR
JOHNSON KINDNESS^{PLLC}

DRAFT

Maria L.C. Anderson
Registration No. 40,574
Direct Dial No. 206.695.1714

On behalf of:

Mauricio A. Uribe
Registration No. 46,206
Direct Dial No. 206.695.1728

MAU:mmm